Atty. Docket No. CH920000003US!
(590.058)

04. 17. 04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of : Bergan et al.

Serial No. : 09/888,698   Examiner : G. Davis

Filed : June 24, 2001   Art Unit : 2121

For : FRACTAL SEMANTIC NETWORK GENERATOR

June 16, 2004

CLAIM FOR PRIORITY UNDER 35 U.S.C. § 119

RECEIVED

JUN 2 2 2004

Technology Center 2100

Assistant Commissioner for Patents:
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

A claim for priority was previously made under the provisions of 35 U.S.C. § 119

for the above-identified U.S. patent application based upon European patent application

No. 00113437.8 filed June 24, 2000. A certified copy of this European patent application

is filed herewith.

Respectfully submitted,
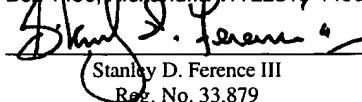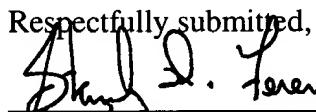
_____
Stanley D. Ference III
Registration No. 33,879

Customer No. 35195
FERENCE & ASSOCIATES
400 Broad Street
Pittsburgh, Pennsylvania 15143
(412) 741-8400
(412) 741-9292 - Facsimile
Attorneys for Applicant

Enclosure

This Page Blank (uspto)

Europäisches Patentamt	European Patent Office	Office européen des brevets

Bescheinigung   Certificate   Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.	The attached documents are exact copies of the European patent application described on the following page, as originally filed.	Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr.   Patent application No.   Demande de brevet n°

00113437.8

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE,   21/03/01
LA HAYE, LE

EPA/EPO/OEB Form   1014   - 02.91

This Page Blank (uspto)

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:   **00113437.8**
Demande n°:

Anmeldetag:
Date of filing: **24/06/00**
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
International Business Machines Corporation
Armonk, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Fractal semantic network generator

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

FRACTAL SEMANTIC NETWORK GENERATOR

The present invention concerns the processing of an input string or input text in order to generate an input network and in particular a fractal semantic input network.

5

BACKGROUND OF THE INVENTION

Natural language understanding is a topic that is attracting a lot of attention since it eventually will allow to improve the interface and interaction between humans and
10   machines, such as computers, personal digital assistants (PDAs), or cellular phones, for example. Natural Language Processing (NLP) is vital in natural language interfaces, machine translation, automatic abstracting and in a number of other computer-based applications.

15   Despite a lot of efforts, the advances in natural language understanding are not very satisfying. In particular, the research area of automated apprehension of meaning from speech or from text has not made much progress over the last years.

In the area of automated apprehension of meaning there are two different philosophies, the
20   "holistic" approach and the "componentized" approach. While the latter approach has achieved a great number of detailed results in the individual components such as anaphora resolution and deep syntactic analysis, among many others, it has not yet achieved the goal of combining these individual components to one global and overall solution within which these components can interact. It is even unclear at the moment whether there exists for
25   these components any suitable form of combination that includes feedback mechanisms.

When it comes to understanding the meaning of speech, i.e. the semantical interpretation of speech, the breakthrough has not yet been achieved. As a consequence, the pragmatical analysis, the control of tools and devices by natural speech, has also not been developed
30   very far. A typical example of a modern speech/text recognition system is described in the article "Enabling agents to work together", by R.V. Guha et al., Communications of the ACM, Vol. 37, No. 7, July 1994, pp. 127-142, and reviewed by T.J. Schult in the German

article "Transparente Trivialitäten; Cyc-Wissensbasis in WWW", c't, 1996, Vol. 10, pp. 118-121. The Cyc-system described by R.V. Guha is a knowledge based system for true/false categorization of input statements. T.J. Schult points out in his article that the knowledge representation in the database used in the Cyc-system is not standardized and

5   uses only the following relations for deduction: 'is element of', 'is subset of', and 'has subsets'. The system described by Guha is what we call a "holistic" system.

In the present context, we focus on the „holistic" approach that is also referred to as cognitive approach. A fractal semantic knowledge database is employed in order to be able

10   to perform a meaning understanding task. This kind of an approach has been used by others before, but the present model to represent the world knowledge in a knowledge database modeled as a fractal semantic network is unique and has a number of important differences compared to all other models around. In particular, a self-similar hierarchical network of n-th order is employed, as for example disclosed and claimed in PCT Patent

15   application WO 99/63455, International Application No.: PCT/IB99/00231, entitled "Processing of Textual Information and Automated Apprehension of Information", currently assigned to the assignee of the present application. Furthermore, local pervasive intelligence is employed, as specified in the German Patent application "Fraktales Netz n-ter Ordnung zum Behandeln komplexer Strukturen", application No.: 199008204.9, filing

20   date 15 February 1999, assigned to the assignee of the present application and to Delphi Creative Technologies. This local pervasive intelligence is called a Janus, using a name of the Roman mythology (plural: Jani). The Janus is described in very general terms in this German patent application. These two patent applications are incorporated in their entirety.

25   It is to be noted that there is a fundamental difference between data or information and knowledge. One can accumulate arbitrary amounts of data or information without having any knowledge, while the converse is not possible. Data or information is the isolated representation of pure facts, while knowledge arises from strong connections between these facts, from connections between facts and their environment, and from abstraction,

30   which in turn allows both for performing understanding and learning.

CH9-2000-0003

3

An approach for the meaning understanding based on a fractal semantic knowledge base is described and claimed in co-pending patent application with title " MEANING UNDERSTANDING BY MEANS OF LOCAL PERVASIVE INTELLIGENCE ". This patent application was filed on the same day and is currently assigned to the same

5   applicant as the instant patent application. This patent application is incorporated in its entirety. According to this co-pending case, local pervasive intelligence (realized by means of Janus objects) is employed in order to process an input network. During the processing of this input network, knowledge is extracted from the knowledge base. This enables such a system to automatically apprehend (understand) what was conveyed in the input network.

10   Before such a meaning understanding task can be carried out, some preparational work is required.

Linguists and programmers have developed and are developing parsers that are able to parse subsets of a language. So far, modern linguistic theories did not lead to parser

15   implementations that have enough lexical information to be able to parse a substantial subset of the English language, for instance. This is due to the fact that linguistic theories cannot deal with all exceptions that a natural language contains, and, therefore, the parsing sometimes fails, generating either wrong or corrupt outputs.

20   One parser giving good results is the English Slot Grammar (ESG) parser developed by Michael McCord of International Business Machines Corporation.

It would be desirable to provide an efficient scheme for the conversion of an input string or input text into an input network suited for meaning understanding.

25

It would also be desirable to provide a system for the efficient conversion of an input string or input text into an input network suited for meaning understanding.

CH9-2000-0003

4

SUMMARY OF THE INVENTION

The present invention concerns the conversion of an input string or input text into an input network that has a fractal semantic structure.

5   The present invention further concerns a specific fractal semantic input network for use in connection with the automated apprehension, interpretation or meaning understanding of information conveyed in an input string.

A method, system, and computer program code for processing an input string to generate a
10   fractal semantic network are disclosed. The fractal semantic network is obtained by parsing the input string to generate an initial parse structure which comprises a plurality of parse elements. Semantic units are generated from some of the parse elements by employing a set of mapping rules for mapping certain types of parse elements into the semantic units such that some of the semantic units are information objects and some are connection
15   objects. The fractal semantic network is generated by arranging the information objects and connection objects in a fractal manner, where the relation between the information objects is represented by the connection objects. The information objects and the connection objects are derived from a ~~small~~ set of basic building blocks.

20   The present invention can be used for meaning understanding in natural language, but also for speech analysis, language translation, information retrieval and other applications which require some sort of natural language processing.

Advantages of the present invention are addressed in connection with the detailed
25   description or are apparent from the description.

CH9-2000-0003

5

DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the invention will now be described with reference to the
following schematic drawings.

5

FIG. 1A   shows a schematic block diagram of a meaning understanding system,
according to the co-pending patent application filed on the same day as the
instant patent application.

FIG. 1B   shows a schematic block diagram of a meaning understanding system,
according to the above-mentioned PCT Patent application WO 99/63455.

10

FIG. 2A –2B   show the various elements (semantic units) of a fractal semantic knowledge
database. This representation is a generalization of what is described in the
above-mentioned in PCT Patent application WO 99/63455.

15   **FIG. 3**   shows a schematic block diagram of a processing system, according to the
present invention.

FIG. 4   shows part of a fractal semantic input network, according to the present
invention.

20

FIG. 5A   shows a screen shot of the parse structure generated by the ESG parser.

FIG. 5B   shows a graphical representation of a fractal semantic input network
generated by a system in accordance with the present invention.

25

FIG. 6A-6B   show a 'pseudo code' representation of a semantic network generator,
according to the present invention, implemented in software.

30

CH9-2000-0003

6

DESCRIPTION OF PREFERRED EMBODIMENTS

In the following, the basic concept of the present invention is described. Before addressing different embodiments, the relevant terms and expressions are defined and explained.

5

The expression "textual information" is defined to be any kind of written information, or information contained in speech. "Textual information" is meant to represent natural language. This expression is also meant to represent natural language transmitted through a network or phone system, and the like.

10

The term "input string" is used as a synonym for any kind of input that can be processed by a software-based computer system. It can be a continuous structured or unstructured string of bits, for example. It can also be some other representation of information, e.g., a frequency spectrum as generated by a speech processing system. The term "input string" is

15   also meant to stand for a plurality of parallel input strings suited for parallel processing.

As illustrated in Figure 1A, there is some sort of an input string 12 that needs to be processed. Such an input string 12 may be generated by a speech recognition system that automatically turns the speech input into a string 12. An input network 18 has to be

20   generated from the input string 12 since the meaning understanding system 10, as disclosed in the co-pending patent application that is filed on the same day as the instant patent application, requires such a network 18 as a starting point. According to this co-pending patent application, the processing of the input network 18 is carried out by special Janus objects. There is a knowledge database 11 that is examined when processing the input

25   network 18, whereby knowledge in this knowledge database 11 is used for meaning understanding.

Another setup is described in the PCT Patent application WO 99/63455. A schematic representation of one such setup is illustrated in Figure 1B. According to this PCT

30   application, a meaning understanding system 10 may comprise the units depicted in Figure 1B. An input string 12 is in this embodiment processed by a semantic preprocessor 17 that

CH9-2000-0003

7

converts the string 12 into an input network 18. The semantic processor 19 then uses the input network 18 as input.

For the purpose of the present description we call the module that converts an input string
5   12 into an input network 18 a "fractal semantic network generator" 17 (or semantic network generator). The input network 18, according to the present invention, is a first guess of the input string's semantic units (e.g., objects, attributes, and their relation). That is, the conversion performed by the fractal semantic network generator 17 structures the input string 12. The input network 18 also contains grammatical information such as tense,
10   case, declination, and so forth.

The term **network** is herein used to describe a collection or aggregation of entries/elements. These entries/elements are – at least to some extent – connected/linked by pointers or connection objects. A simple example of a network (input network 18) is given
15   in Figure 4. In this example, the network 18 comprises five objects 50.1 – 50.5 and four pointers 51.1 – 51.4 (i.e., nine semantic units).

We continue with the basic definitions of semantic and hierarchical networks.

20   A **semantic network** is a directed or non-directed graph with the additional property that its nodes and pointers carry semantic information. Nodes and pointers of a semantic network are called semantic units. The semantic information provides a particular meaning for a semantic unit. Frequently, semantic information is encoded in natural language, i.e. specific names or processes are used to label semantic units. _
25

A **hierarchical network** is a directed or non-directed graph with the additional property that some of its pointers carry scaling information. This means that one of the nodes connected to a scaling pointer is at a higher level of hierarchy than the other node.

30   It should be noted that this definition does not yield an absolute value for the level of hierarchy, i.e. it does not assign to every node an integer that corresponds to its level of hierarchy. Instead it gives a relative definition for the level of hierarchy.

CH9-2000-0003

8

The knowledge database 11 is of hierarchical and semantical nature and all elements are alike, so that the knowledge database 11 has a fractal structure, i.e., it is self-similar across hierarchies. Algorithms can operate on elements at any hierarchical level in the same way,

5　　making them 'fractal' algorithms.

This knowledge database 11 may have a unique structure as for example described in the PCT Patent application WO 99/63455. All elements and aspects of the knowledge database 11 that are relevant for the understanding of the present invention are addressed below.

10

Knowledge database: A knowledge database is a kind of library describing the knowledge of the world, or a particular area of interest thereof, by using a fractal semantic network structure.

15　　When referring to a knowledge database, either a library describing knowledge of the world, or a library with domain specific information is meant, or a mixture of both. An example of a domain specific knowledge database is a database which comprises knowledge relevant for the processing of insurance claims.

20　　Self-similar local algorithms can be used when making use of the knowledge contained in this knowledge database.

According to the present invention, the input network 18, like the knowledge database, is of hierarchical and semantical nature and all its elements are alike.　　　——

25

Semantic units: A semantic unit is a set that contains one or several pieces of information. It may be represented by a word, an object, a relation, an attribute, a combination of words and/or objects and/or relations and/or attributes and/or modules, a (hierarchical) network of words and/or objects and/or relations and/or attributes and/or modules, a part of a sentence

30　　or a whole sentence, a part of a paragraph or a whole paragraph, or a part of a story or a whole story.

CH9-2000-0003

9

Exemplary types of semantic units and their possible mutual connections, are schematically illustrated in Figures 2A and 2B. The input network 18 may comprise the semantic units illustrated in these Figures.

5　　A simple example of an input network 18 with semantic units 50.1 – 50.5 and 51.1 – 51.4 is illustrated in Figure 4. The objects 50.1 – 50.5 are represented by circles and the pointers by lines 51.1 - 51.4.

A **higher-order network** is a directed or non-directed graph in which pointers can at the

10　　same time be nodes. This means that a pointer can connect two nodes, one node and one pointer, or two pointers.

In the next definitions it is captured what is meant by a fractal network. A working definition is given which will suffice for the scope of the present description. In particular,

15　　the presently used formulation allows us more easily to define the self-organizing fractal semantic network and to understand the various embodiments and implementation details given.

A **fractal network** is a hierarchical network with the additional property that all of its

20　　nodes and pointers are derived from a small set of basic building blocks. In this sense a fractal network exhibits a self-similar structure because it looks the same everywhere on all levels of hierarchy.

As specified above, nodes and pointers of a network are called semantic units (cf. Figures

25　　2A and 2B). All semantic units 100 are subdivided into concepts and instances. We further subdivide nodes into information units 101, attribute units 102, and processing units or Janus Objects 103. Information units are general elements that can represent concepts or instances, and they are identified by specific names. Attribute units 102 are identified by specific names and values, which can be set, retrieved, or computed.

30

All pointers of the network (connection objects 104) are either scaling (hierarchical connections 105) or non-scaling scaling (non-hierarchical connections 106). Standard

CH9-2000-0003

10

inheritance principles are defined across all scaling pointers, making use of the network's topology or neighborhood concept. The notion of topology and the neighborhood concept are addressed in the co-pending patent application that was filed on the same day as the instant application. Pointers are further subdivided into comparison units 107, 109,

5   interaction units 108, 110, description units 111, role units 112, and controller units 113. Non-scaling comparison units 109 allow us to describe the degree of similarity or dissimilarity of two semantic units, while scaling comparison units 107 allow us to describe how close one semantic unit comes to being an instance of another semantic unit, or how close one semantic unit comes to being a special case of another semantic unit.

10   Non-scaling interaction units 110 allow us to describe the type of interaction of two semantic units, while scaling interaction units 108 allow us to describe the 'role' one semantic unit plays with respect to another semantic unit. Description units connect semantic units to their attribute units, which describe the semantic units in more detail. Role units describe the role one semantic unit plays with respect to another semantic unit.

15   Finally, controller units connect semantic units to their Janus Objects 103, which in turn control and act upon the semantic units' local neighborhoods.

In accordance with the above definition, a network is called a fractal semantic network, if the following four conditions are satisfied:

20   • All nodes are similar (derived from one template) across all hierarchies.
         • All pointers are similar (derived from one template) across all hierarchies.
         • Pointers may also be nodes across all hierarchies.
         • Hierarchical pointers are possible, and at least one node must have a hierarchical pointer.

25   The construction of a fractal semantic network can be achieved as follows. The network is given by a list of semantic units 100 and pointers (connection objects 104), as illustrated in Figures 2A and 2B. There might be different types of semantic units (as defined above) and pointers (similarity pointers 107, 109, functional pointers 108, 110, attribute pointers 111, and role pointers 112, also defined above). These pointers can be oriented upwards,

30   downwards, or horizontally. The various semantic units 100 are interconnected in various

CH9-2000-0003

11

ways. Some of the pointers 105 are hierarchical representing the multiscale approach. Knowledge can thus be presented in the input network 18 as an associative network.

5　A semantic unit may be linked by associative pointers to its associative semantic unit(s). These associative pointers reflect how one understands each semantic unit. It is emphasized that these pointers can exist between any two semantic units. The pointers themselves may be regarded as semantic units that can have pointers to other semantic units, reflecting the fact that something could act on the association between two semantic units rather than on the individual semantic units. Due to the fractal organization of the

10　input network 18, as outlined above, an input string can be significantly restructured. The well defined fractal structure of the input network 18 greatly simplifies the task of trying to understand the information that is mapped from the input string 12 into the input network 18, as will be addressed now.

15　Extracting knowledge from the knowledge database 11 in order to enrich the information originally contained in an input network 18 can be done in a very efficient manner due to the fact that the input network has a fractal semantic structure. Using a local and iterative approach – as described in the co-pending patent application that was filed on the same day as the instant application - one can use the knowledge contained in the knowledge database

20　11 to perform the understanding task of what is conveyed in the input string 12. The key is that this is done locally, making use of the fractal structure of the knowledge database 11, triggered by elements of the input network 18.

　The herein proposed schemes and systems are based on the above-described special model

25　of textual information and natural language. According to this model, natural language as well as textual information consists of semantic units which are grouped at different levels of hierarchy and are all of a similar type.

　According to the present invention, it is advantageous to generate an input network 18 out

30　of an input string 12 that has a fractal semantic network structure like the knowledge database 11. Preferably, the knowledge database 11 and the input network 18 are all represented in what is called a higher-order fractal semantic network. The input network 18

CH9-2000-0003

12

of a speech input 12 (e.g., a string representing question) may be created by locating its semantic units, possible relations, possible attributes, and possible roles. A meaning understanding scheme may then be employed to generate a meaningful output 15.

5   It is advantageous for a subsequent meaning understanding process that the structure of the input network 18 and the knowledge database 11 are similar, since such a subsequent meaning understanding process attempts to bridge or merge the input network 18 and portions of the knowledge database 11.

10   As illustrated in Figure 3, a parser 8, according to the present invention, is employed that transforms the input string 12 into an initial network 9. This initial network 9 is referred to as initial parse structure. It may be a tree-structured network 21, as illustrated in Figure 3, or an appropriate other formalistic representation 50, as illustrated in Figure 5A. The semantic network generator 17 further comprises a processing module 7 that converts the

15   initial network 9 into the desired input network 18.

Even without a parser a hierarchical network may be generated as long as there is some unit that parses the input string in order to generate parse elements. The parsing in this case can be a simple mapping.

20

In the present embodiment, the fractal semantic network generator 17 comprises a parser 8 that generates a tree-structured network 21. The whole conversion from a string of text 20 (e.g. one single sentence) via the tree-structured network 21 to a fractal semantic network 22 is schematically illustrated on the right hand side of Figure 3.

25

Preferably, an ESG parser is employed. Details of this ESG parser are addressed in the following two exemplary publications: McCord, M. (1990), "A System for Simpler Construction of Practical Natural Language Grammars", R. Studer (ed.), Natural Language and Logic, Lecture Notes in Artificial Intelligence 459, London and New York, Longman;

30   and Lappin, S. & M. McCord (1990), "Anaphora Resolution in Slot Grammar", Computational Linguistics, 16(4), pp.197-212.

CH9-2000-0003

13

There is also an IBM US patent with patent No. US 5,737,617, which addresses certain aspects of McCord's work, in particular a method and system for English text analysis.

Slot grammar is a formalistic representation of information conveyed in input strings. It
5   allows this information to be represented in a manner (herein called parse structure) that can be processed by a computer system. The ESG parser is capable of processing many of the phenomena of a natural language, such as the English language. The development of slot grammar parsers for languages other than English is in progress.

10   The ESG parser delivers parse structures showing both surface structure and deep structure. Preferably, the parser produces a parse structure that resembles a tree structure (e.g. the tree structure 21 of Figure 3). Usually, the parser delivers one parse structure per input sentence and preferably one single tree per input sentence 20. The ESG parser processes an input string 12 and yields a parse structure with slot information for each
15   sentence of the input string 12. In addition, a variety of miscellaneous syntactic and semantic information is added to each word of the input string 12.

A typical parse structure 50 with miscellaneous syntactic and semantic information is illustrated in Figure 5A. Figure 5A is a screenshot that is generated by the ESG parser after
20   having processed the following sentence (input string 12):

<<I approached an intersection with no traffic lights and no signs.>>

The first column 51 of the parse structure 50 represents the relationship between the
25   various parse elements in this sentence. The second column 52 shows the semantic information of all parse elements in this sentence (the parse element characterizes a word, or a phrase, and describes its meaning) is used . The parse element can even be a comma or the like. The word "I" is for example characterized as the subject of the sentence. In the third column 53, the word stem, the meaning, and the slot information is given. The
30   number 1 before the parenthesis in the expression "approach1(2,1,4,u)" means the first meaning of the word "approach" is used in the sentence. The lexical information used by the parser indicates that there are several different meanings for the word "approach".

CH9-2000-0003

14

Within the parenthesis the 2 refers to the second word of the sentence, that is, it indicates that the word "approach" is related to the first word "I". The number 4 indicates that the subject "I" "approach" the "intersection" (the fourth word in the sentence). The lexical information of the ESG parser knows for many words the corresponding slot information.

5　Whenever the word "approach" appears as a verb in a sentence, then the parser tries to locate the subject ("I") and the object ("intersection"). The fourth digit in the expression "approach1(2,1,4,u)" shows a "u" which stand for "unknown" and is related to different usage of "approach" for which still another slot may be filled. The last column 54 gives feature information about features of each word in the sentence to the extent known by the

10　parser. I.e., this column represents and describes the syntax of the sentence. The word "I" is identified as being a noun, a pronoun, a nominative form, and the 1st person. This syntactic information is the result of a deep syntactic analysis followed by a shallow semantic analysis of the sentence. It is obvious that what is shown in Figure 5A (the parse structure 50) can be represented in many different formats.

15

In general terms, the ESG parser returns semantic information, syntactic information, information concerning the relationship/association between some or all of the words of the sentence, and feature information. In addition, the ESG parser returns the individual words of the sentence.

20

According to the present invention, a fractal semantic network generator 17 is defined that creates a fractal network of semantic units (referred to as a fractal semantic network), where the semantic units are derived from the individual words, from the semantic information, and from the syntactic information. In addition, the present semantic network

25　generator 17 also processes the information concerning the relationship/association (e.g., the information given in the first column 51 and in the third column 53 of the screenshot 50) between some or all of the words, such as the tree structure provided by the ESG parser.

30　A schematic representation of the basic software components (in pseudo code format) of one embodiment of the present invention are illustrated in Figures 6A and 6B. The first software component 60 shows the mapping of information 61 provided by the ESG parser

CH9-2000-0003

15

into corresponding semantic units 62. In the present embodiment, nouns, verbs, qualifiers, subordinary conjunctions, and prepositions are mapped into information objects, whereas determiners, adverbs, adjectives, and infinitive to are mapped into attribute objects.

5   The second software component 63 deals with the conversion of information taken from the parse structure into connection objects. The present embodiment has nine different modules $(1 - 9)$ for this conversion. The pseudo code is to be read like an instruction. The sign "¦" means OR, the sign "<>" means NOT EQUAL, and the sign "=" means EQUAL. One has to handle/process two words (word and otherWord in the present example). Every

10   variable that starts with the word "other" refers to the other word (the second word), or else the variable refers to the original word (first word). The first module 1) checks whether a verb (element 0) has been identified in the parser's syntactic information (e.g. column 54 in Figure 5A). The module then looks at the slot information of this verb (slot of element 0) to find another word's position in the sentence. If this other word is a noun, which can

15   be verified by looking at this other word's syntactic information in the parse structure 50, then an agent role connection is generated in the input network 18. The second module 2) generates an object role connection, the third module 3) an attribute connection, the fourth module 4) a scaling exchange connection, the fifth module 5) a horizontal relation, the sixth module 6) a special horizontal relation, the seventh module 7) yet another special

20   horizontal relation, the eighth module 8) an agent role connection, and the ninth module 9) a beneficiary role connection. Else unspecified connections are created.

According to one particular embodiment of the present invention, the above rules can be reformulated if that is deemed convenient.

25

The two software modules 60 and 63 process the information received from the initial network 9 in a manner illustrated in Figure 7. In a first step, a unit 23 decides whether the received information is to be fed to the first software module 60 or the second module 63. This can be done by trying to make a distinction between semantic units that have to be

30   processed by module 60 and semantic units that have to be processed by the module 63. A possible implementation is to check whether the information received requires a semantic unit decision. If the answer is yes the information is forwarded to module 60.

CH9-2000-0003

16

Else, the information is forwarded to the module 63. Optionally, there may also be an exchange of information between the two modules 60 and 63, as indicated by the arrow 24.

5   These two modules 60 and 63 feed another module 64 that generates the fractal semantic input network 18 out of the information provided to it by the modules 60 and 63. For this purpose, the module 64 arranges the information accordingly. It either comprises a memory or it has access to an external memory (e.g. a disk drive) where the input network 18 is generated. This is done by storing the semantic units, by linking the semantic units, and so forth.

10

Using the first software component 60, the second software component 63, and the module 64 (which in most implementations also is realized as software) one can transform the initial network 9 into a fractal semantic network 18. Still referring to the single sentence given above, the resulting fractal semantic input network 18 is illustrated in Figure 5B. As

15   discussed in connection with the word "approach", there is somebody or something that approaches somebody or something else. Here, the sentence states that "I approach intersection". This is represented in the input network 18 by the semantic units 70 – 74. The sentence further reveals some specifics about the "intersection". The fractal semantic network generator 17 is able to recognize that the preposition "with" is a bridge between

20   the word "intersection" and the partial sentence "no traffic lights and no signs". The generator 17 notices (understands) that the "intersection" is characterized in that is neither has traffic lights nor signs. All this information is thus placed on a lower level in the fractal semantic network's hierarchy. Scaling exchange connections 75 and 76 (hierarchical connections) are used to link the word "intersection" with the words of this partial

25   sentence.

Note that this is just one implementation example. One can add additional mapping/conversion rules, or one can modify or even omit certain of these rules. It is obvious, that the use of fewer rules may lead to less reliable results. The more exceptions

30   one includes in creating an input network, the more reliable will the resulting input network be.

CH9-2000-0003

17

It is to be noted that some or all of the above processes lead to a logical rearrangement/restructuring of the elements of the initial network 9. The result of the present semantic network generation scheme may not necessarily be reflected in that any physical changes take place inside the initial network 9. The pointers and the like may be
5   pointers in some table or memory space, for example, that become rearranged during the generation of the input network 18.

Note that the input network 18 will never be completely satisfactory. It will always be a naive guess until the knowledge database 11 is confronted.
10

According to a special implementation of the present invention, an exception output (e.g. a file) is generated for those portions of the input string 12 that cannot be parsed into parse elements (e.g. if the parser fails) or for those semantic units that do not seem to fit into the input network 18. The exception output can be processed by a special tool to allow a
15   subsequent parsing into semantic units or a subsequent inclusion in the input network 18. This special tool may allow a user to interact with the system 7. The system 17 may comprise an input device (e.g., a keyboard) for allowing human interaction.

CH9-2000-0003

18

CLAIMS

1. Method for processing an input string to generate a fractal semantic network (18), comprising the steps:

5
- parsing the input string (12) to generate an initial parse structure (9) comprising a plurality of parse elements;

- generating semantic units from some of the parse elements by employing a set of mapping rules for mapping certain types of parse elements into the semantic units, whereby some of the semantic units are information objects and some are connection

10   objects;

- generating the fractal semantic network (18) by arranging the information objects and connection objects in a fractal manner, where the relation between the information objects is represented by the connection objects,

whereby the information objects and the connection objects are derived from a ~~small~~ set of

15 basic building blocks.

2. The method of claim 1, whereby the parse elements comprise a classifier characterizing the type of the parse element, and a representation describing the relation between the parse elements, and feature information describing the features of the parse element.

20

3. The method of claim 1, whereby there are hierarchical connection objects and non-hierarchical connection objects.

4. The method of claim 3, whereby due to the fact that hierarchical connection objects are

25   employed, the fractal network structure has at least two levels of hierarchies.

5. The method of claim 1, whereby the fractal semantic network (18) is self-similar across all hierarchies since the information objects and the connection objects are derived from the ~~small~~ set of basic building blocks.

30

CH9-2000-0003

19

6. The method of claim 1, whereby the input string (12) is a sequence of **m** words, preferably a single sentence.

7. The method of claim 6, whereby **m** semantic units are created during the parsing step, with one semantic unit per word, or per sentence, or per a whole story.

8. The method of claim 2, whereby the representation describing the relation between the semantic units is examined when generating the fractal semantic network (18).

9. The method of claim 1, whereby an exception output is generated for those portions of the input string (12) that cannot be parsed into parse elements or that do not seem to fit into the fractal semantic network (18).

10. The method of claim 9, whereby the exception output can be processed by a ~~special~~ tool to allow a subsequent parsing into parse elements or a subsequent inclusion in the fractal semantic network (18).

11. The method of claim 10, whereby the ~~special~~ tool allows human interaction.

12. The method of claim 1, whereby lexical information is taken into consideration during the parsing step.

13. The method of claim 1, whereby grammar rules are taken into consideration during the parsing step.

14. The method of claim 1, whereby word stemming is applied.

15. The method of claim 1, whereby a syntactic analysis and/or a semantic analysis of the input string (12) is performed during the parsing step.

CH9-2000-0003

20

16. The method of claim 1, whereby the set of mapping rules comprises a first set of rules which define that parse elements representing nouns, verbs, qualifiers, subordinary conjunctions, or prepositions are mapped into the information objects.

5   **17.** The method of claim 1, whereby the set of mapping rules comprises a second set of rules which define that parse elements representing determiners, adverbs, adjectives, and infinitive to are mapped into attribute objects.

18. The method of claim 1, whereby the fractal semantic network (18) is a first guess of the
10   input string's words.

19. The method of claim 1, whereby knowledge is presented in the fractal semantic network (18) as an associative network.

15   **20.** System for processing an input string to generate a fractal semantic network (18), comprising:

• a parser (8) for processing the input string (12) to generate an initial parse structure (9) comprising a plurality of parse elements;

• a processing module (7) for converting the initial parse structure (9) into the fractal
20   semantic network (18), comprising components

—   for generating self-similar information objects from some of the parse elements by employing a first set of mapping rules for mapping certain types of parse elements into the information objects,

—   for generating self-similar connection objects from some of the parse elements by
25   employing a second set of mapping rules for mapping certain types of parse elements into the connection objects; and

—   for arranging the information objects and the connection objects in a fractal manner according to the relation between the respective information objects and connection objects, where the relation between the information objects is represented by the
30   connection objects,

CH9-2000-0003

21

whereby the information objects and the connection objects are derived from a ~~small~~ set of basic building blocks.

21. The system of claim 20, wherein the parse elements comprise a classifier
5　　　　characterizing the type of the parse element, and a representation describing the relation between the parse elements, and feature information describing the features of the parse element.

22. The system of claim 20, wherein the parser (8) is a slot grammar parser.
10

23. The system of claim 20, wherein there are hierarchical connection objects and non-hierarchical connection objects.

24. The system of claim 23, wherein due to the fact that hierarchical connection objects are
15　　　　employed, the fractal network structure (18) has at least two levels of hierarchies.

25. The system of claim 20, wherein the fractal semantic network (18) is self-similar across all hierarchies since the information objects and the connection objects are derived from the ~~small~~ set of basic building blocks.
20

26. The system of claim 20, wherein the system (17) generates an exception output for those portions of the input string (12) that cannot be parsed into parse elements by the parser or for those semantic units that do not seem to fit into the fractal semantic network (18).
25

27. The system of claim 26, further comprising a ~~special~~ tool to allow the subsequent parsing of the exception output into parse elements, or the subsequent insertion of the semantic units into the fractal semantic network (18).

30　　28. The system of claim 27, wherein the ~~special~~ tool enables human interaction with the system (17).

CH9-2000-0003

22

29. The system of claim 20, comprising a memory with lexical information for consideration during the parsing step.

30. The system of claim 20, comprising a memory with grammar rules for consideration
5   during the parsing step.

31. The system of claim 20, wherein the first set of mapping rules is accessible by the processing module (7), and wherein the first set of mapping rules defines that semantic units representing nouns, verbs, subordinary conjunctions, or prepositions are mapped
10   into the information objects.

32. The system of claim 20, wherein the second set of mapping rules is accessible by the processing module (7), and wherein the second set of mapping rules defines that semantic units representing determiners, adverbs, adjectives, and infinitive to are
15   mapped into attribute objects.

33. The system of claim 20, wherein the fractal semantic network (18) is a first guess of the input string's words.

20   34. The system of claim 20, further comprising an output device (64) for providing the fractal semantic network (18) to another system or software module.

35. The system of claim 20, further comprising an input device for allowing human interaction.
25

36. A text processing system comprising a system according to one of the claims 20 – 35.

37. A speech processing system comprising a system according to one of the claims 20 – 35.

30

CH9-2000-0003

23

38. Computer program code for processing an input string to generate a fractal semantic network (18), that, when being executed by a computer system, performs the following steps:

- parsing the input string (12) to generate an initial parse structure (9) comprising a
5   plurality of parse elements;

- generating semantic units from some of the parse elements by employing a set of mapping rules for mapping certain types of parse elements into the semantic units, whereby some of the semantic units are information objects and some are connection objects;

10   - generating the fractal semantic network (18) by arranging the information objects and connection objects in a fractal manner, where the relation between the information objects is represented by the connection objects,

whereby the information objects and the connection objects are derived from a ~~small~~ set of basic building blocks.

15

39. The program code of claim 38, wherein the parse elements comprise a classifier characterizing the type of the parse element, and a representation describing the relation between the parse elements, and feature information describing the features of the parse element.

20

40. The program code of claim 38, wherein there are hierarchical connection objects and non-hierarchical connection objects.

41. The program code of claim 40, wherein due to the fact that hierarchical connection
25   objects are employed, the fractal network structure has at least two levels of hierarchies.

42. The program code of claim 38, wherein an exception output is generated for those portions of the input string (12) that cannot be parsed into parse elements or that do not
30   seem to fit into the fractal semantic network (18).

CH9-2000-0003

24

43. The program code of claim 42, whereby the exception output can be processed by a special tool to allow a subsequent parsing into parse elements or a subsequent inclusion in the fractal semantic network (18).

5   **44.** The program code of claim 43, whereby the special tool allows human interaction.

45. The program code of claim 38, wherein a syntactic analysis and/or a semantic analysis of the input string (12) is performed during the parsing step.

10   **46.** The program code of claim 38, wherein the set of mapping rules comprises a first set of rules which define that parse elements representing nouns, verbs, qualifiers, subordinary conjunctions, or prepositions are mapped into the information objects, and a second set of rules which define that parse elements representing determiners, adverbs, adjectives, and infinitive to are mapped into attribute objects.

15

CH9-2000-0003

25

ABSTRACT

5   Method, system, and computer program code for processing an input string to generate a
fractal semantic network (18). The fractal semantic network (18) is obtained by parsing the
input string (12) to generate an initial parse structure (9) which comprises a plurality of
parse elements. Semantic units are generated from some of the parse elements by
employing a set of mapping rules for mapping certain types of parse elements into the
semantic units such that some of the semantic units are information objects and some are

10 connection objects. The fractal semantic network (18) is generated by arranging the
information objects and connection objects in a fractal manner, where the relation between
the information objects is represented by the connection objects. The information objects
and the connection objects are derived from a ~~small~~ set of basic building blocks.

15

(Fig. 3)

This Page Blank (uspto)

24-06-2000
CH9-2000-0003
EP00113437.8
EPO - Munich
38
2 4. Juni 2000
EPO - Munich
3°
2 4. ......
DRAW
1/10
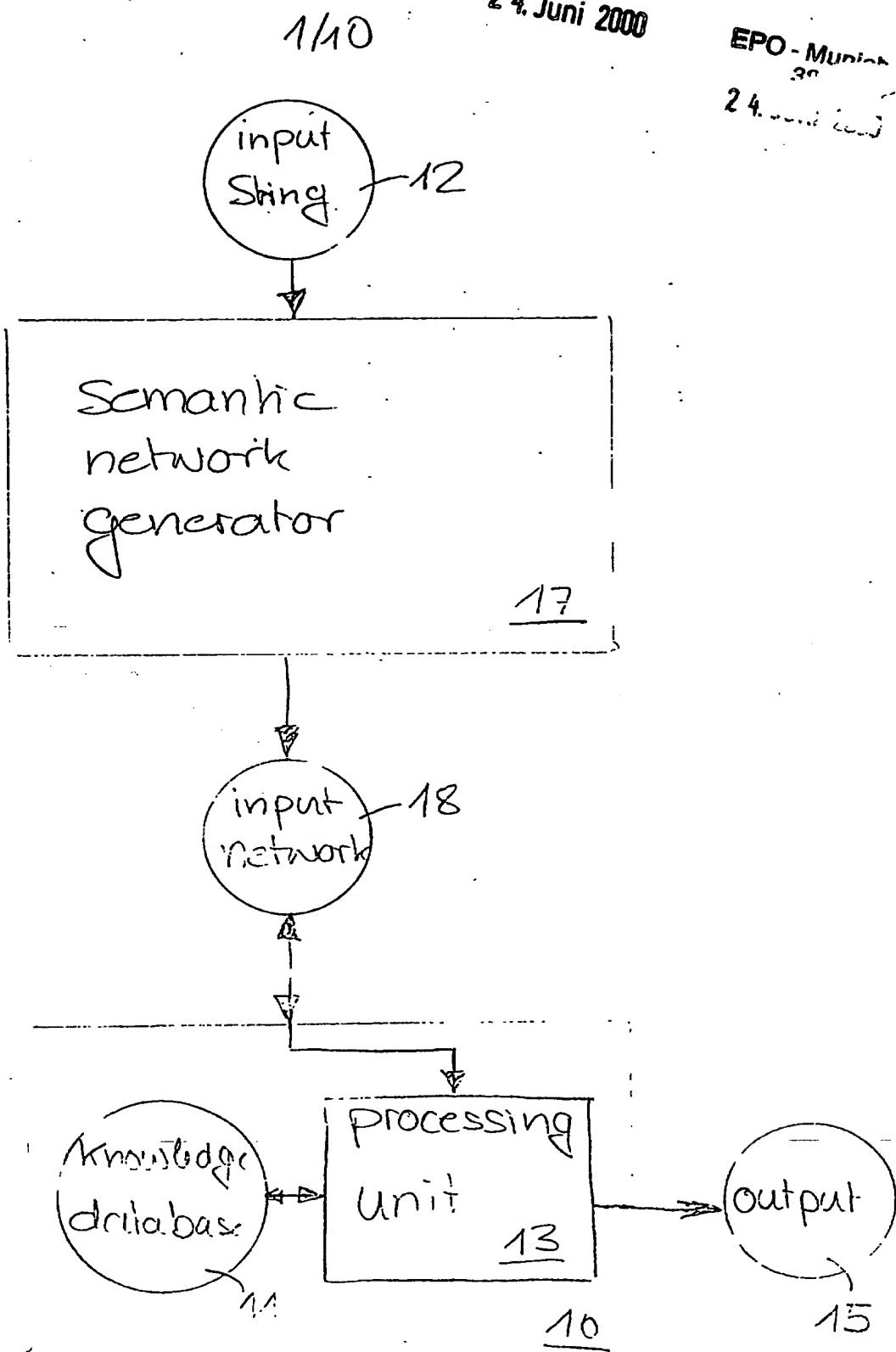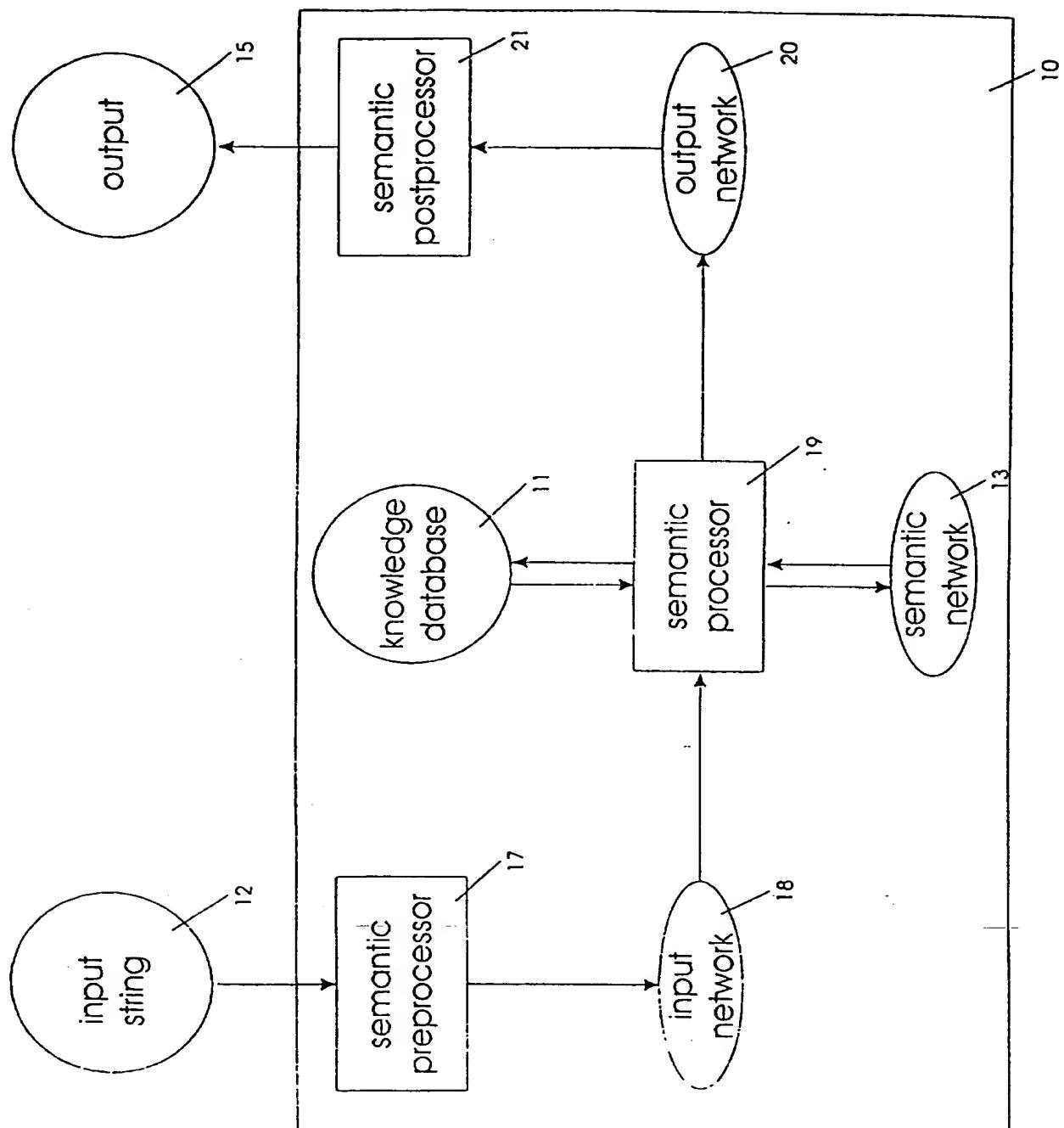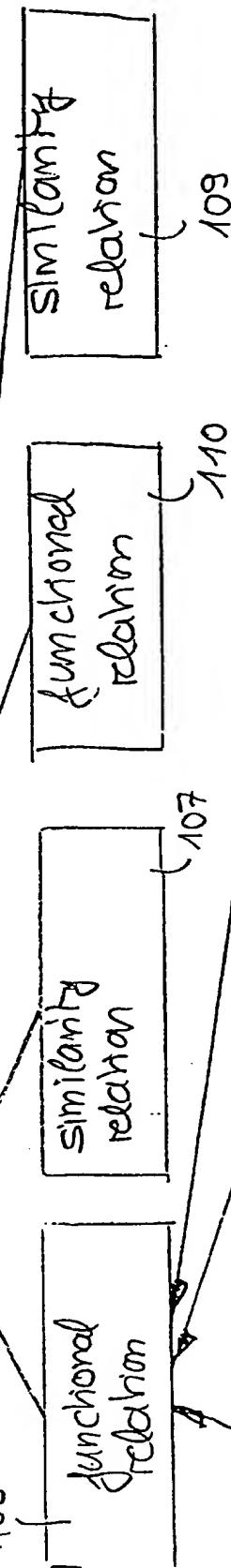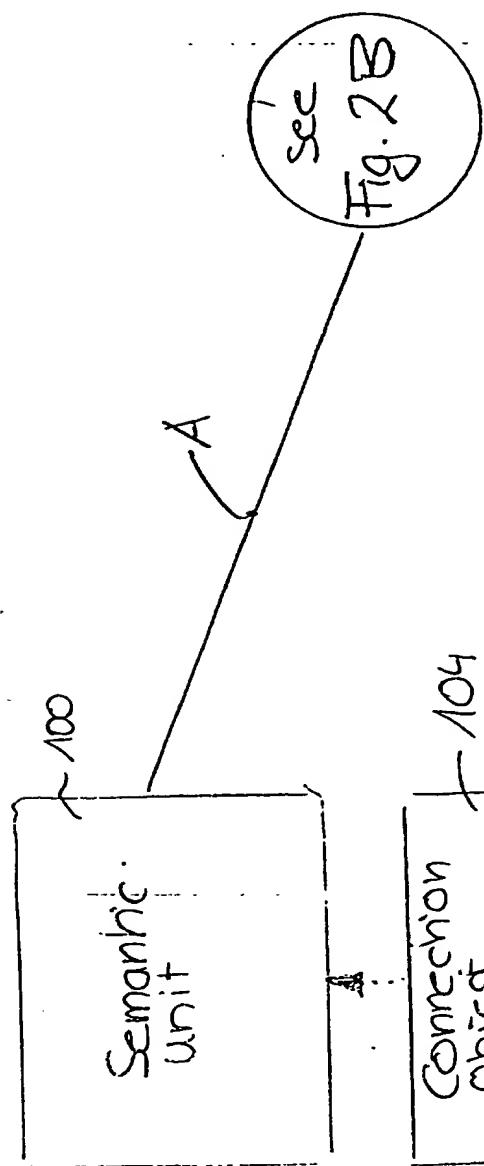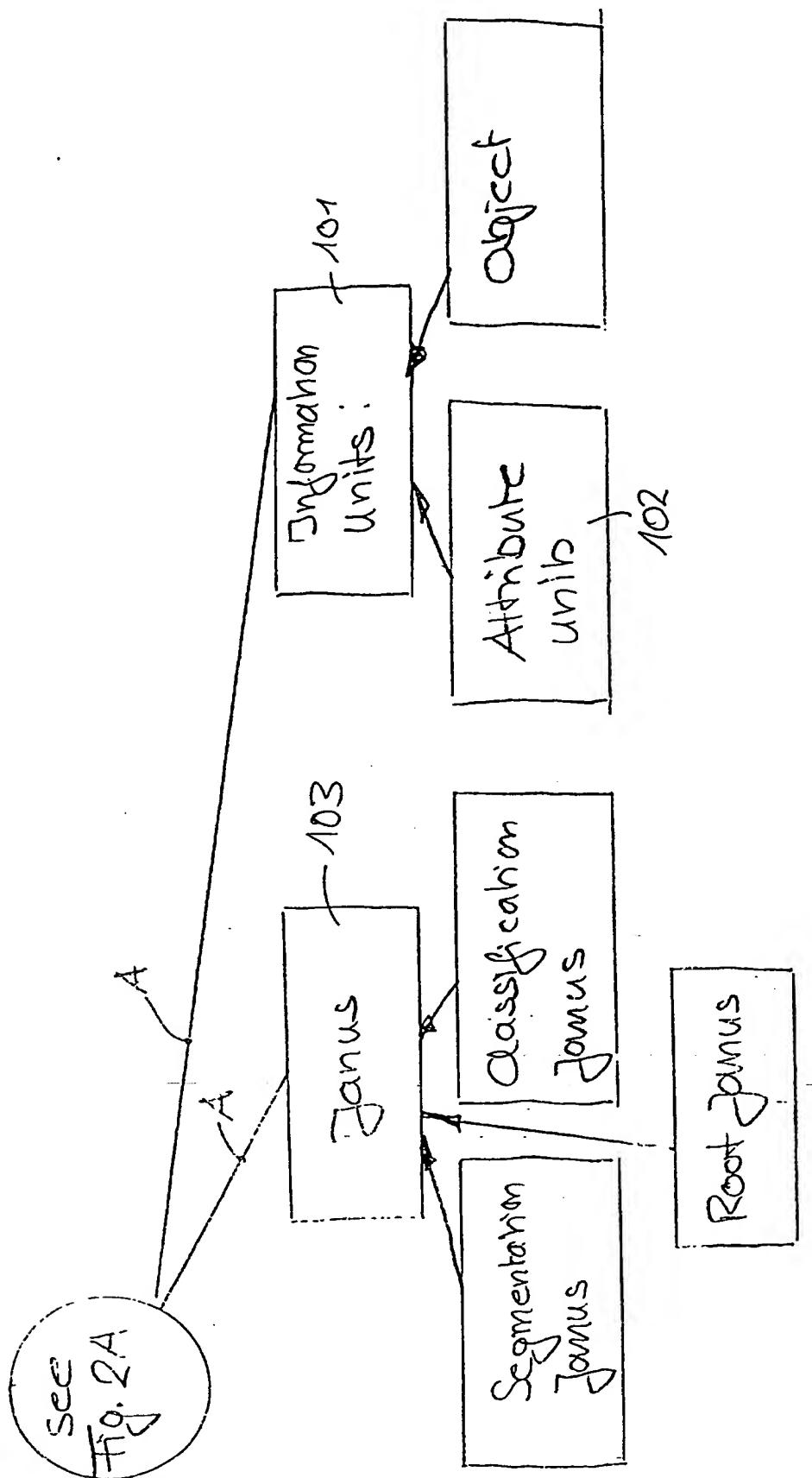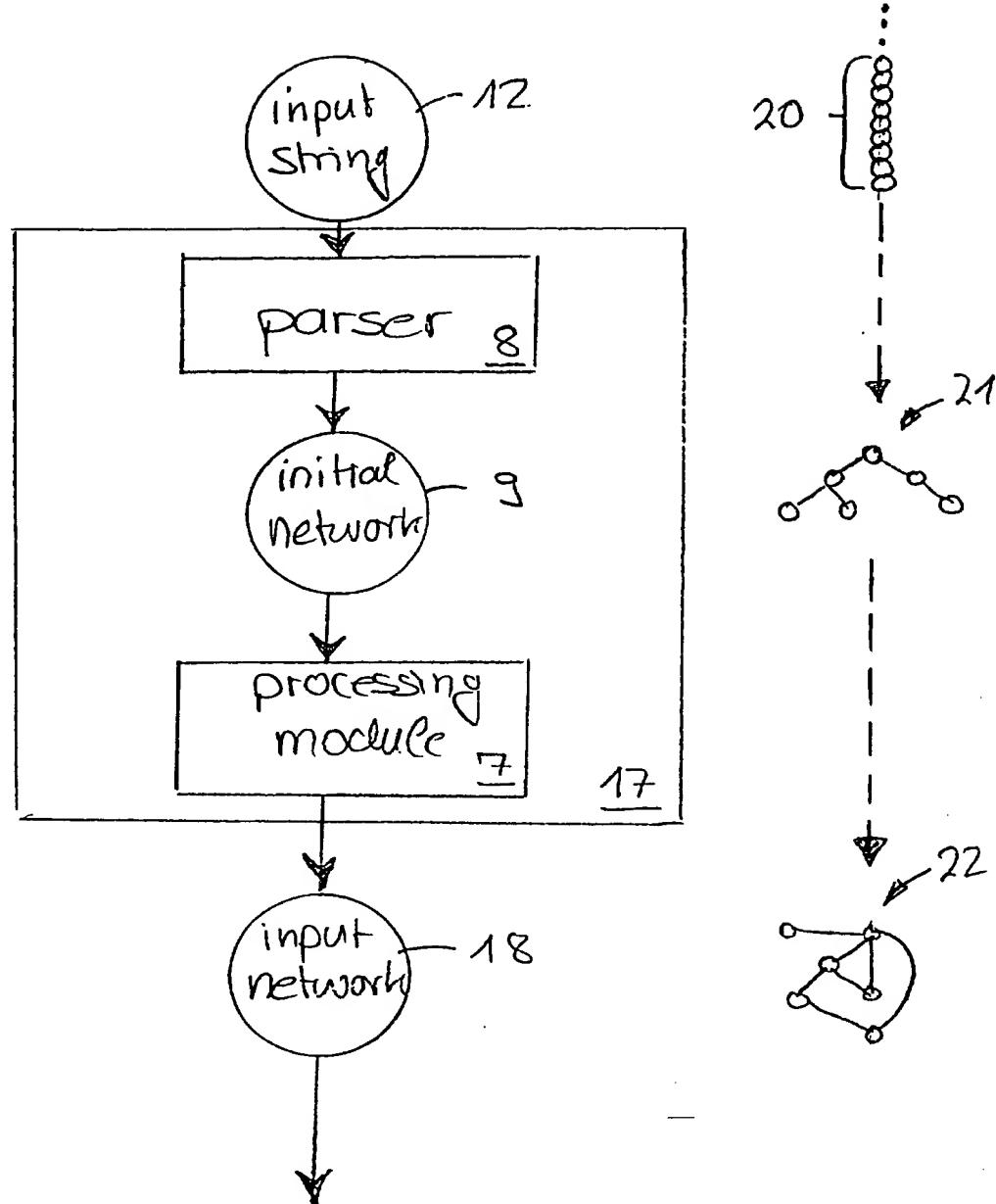
input
String 12

Semantic
network
generator

17

input 18
network

Knowledge
database

processing
unit

13

10

output

15

11

Fig. 1A

2/10

Fig. 1B

Fig. 2A

4/10

Fig. 2B

5/10

Fig. 3

6/10



Fig. 4

Fig. 7

Fig. 5A

Fig 5B
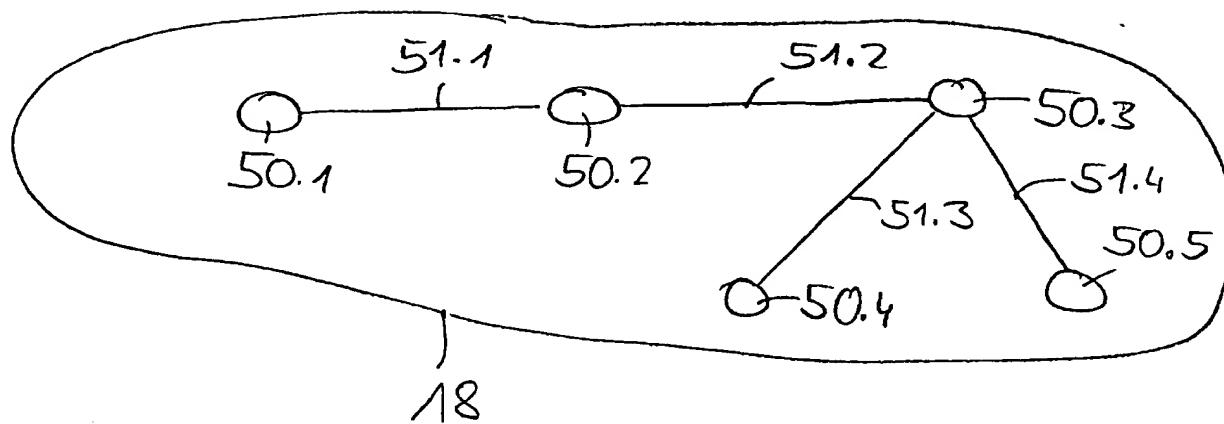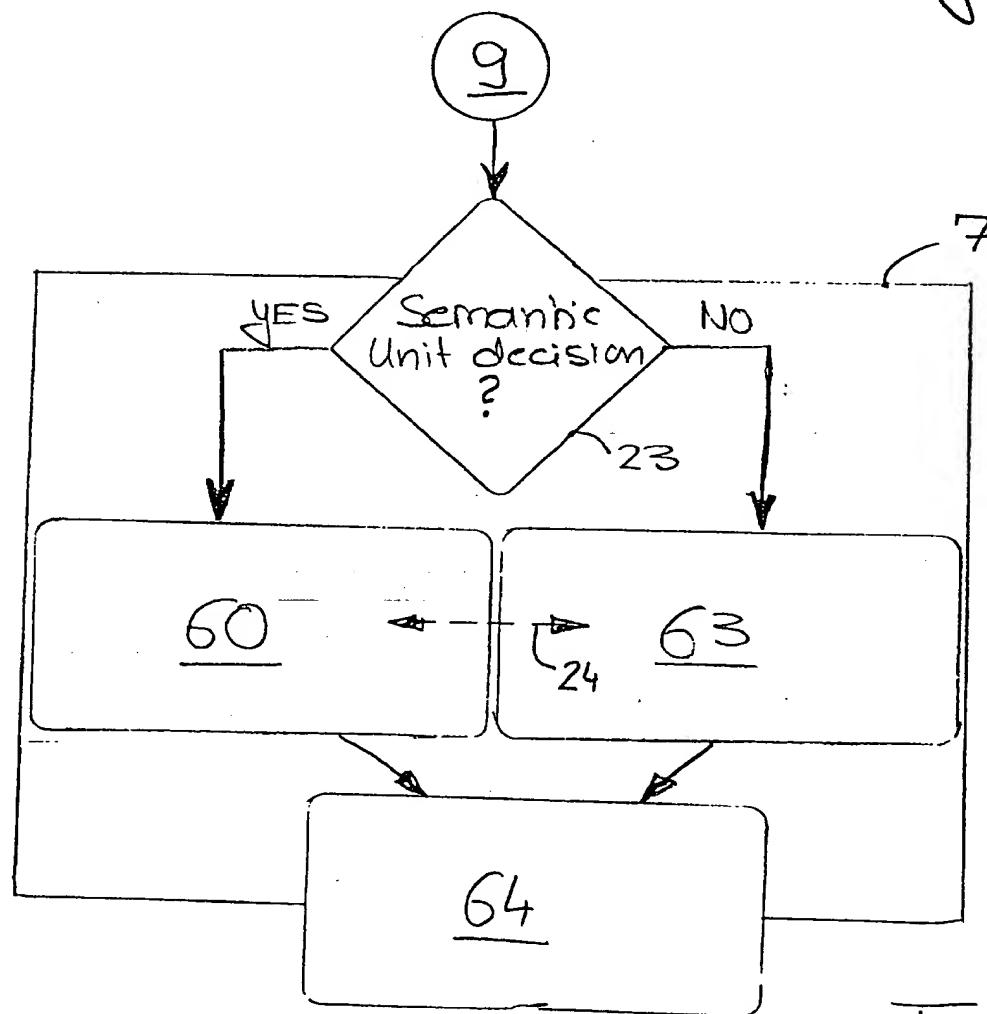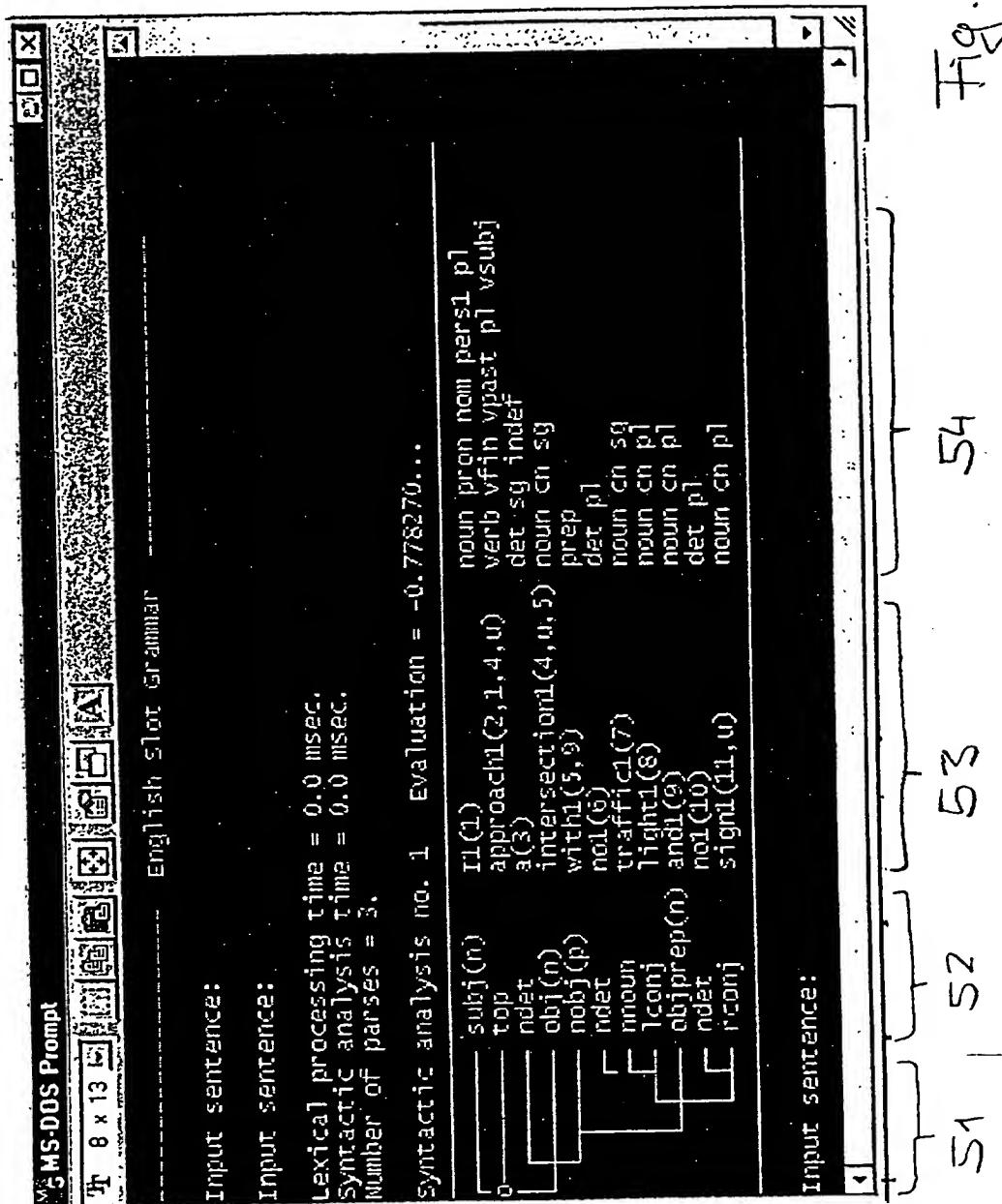
CH9-2000-0003
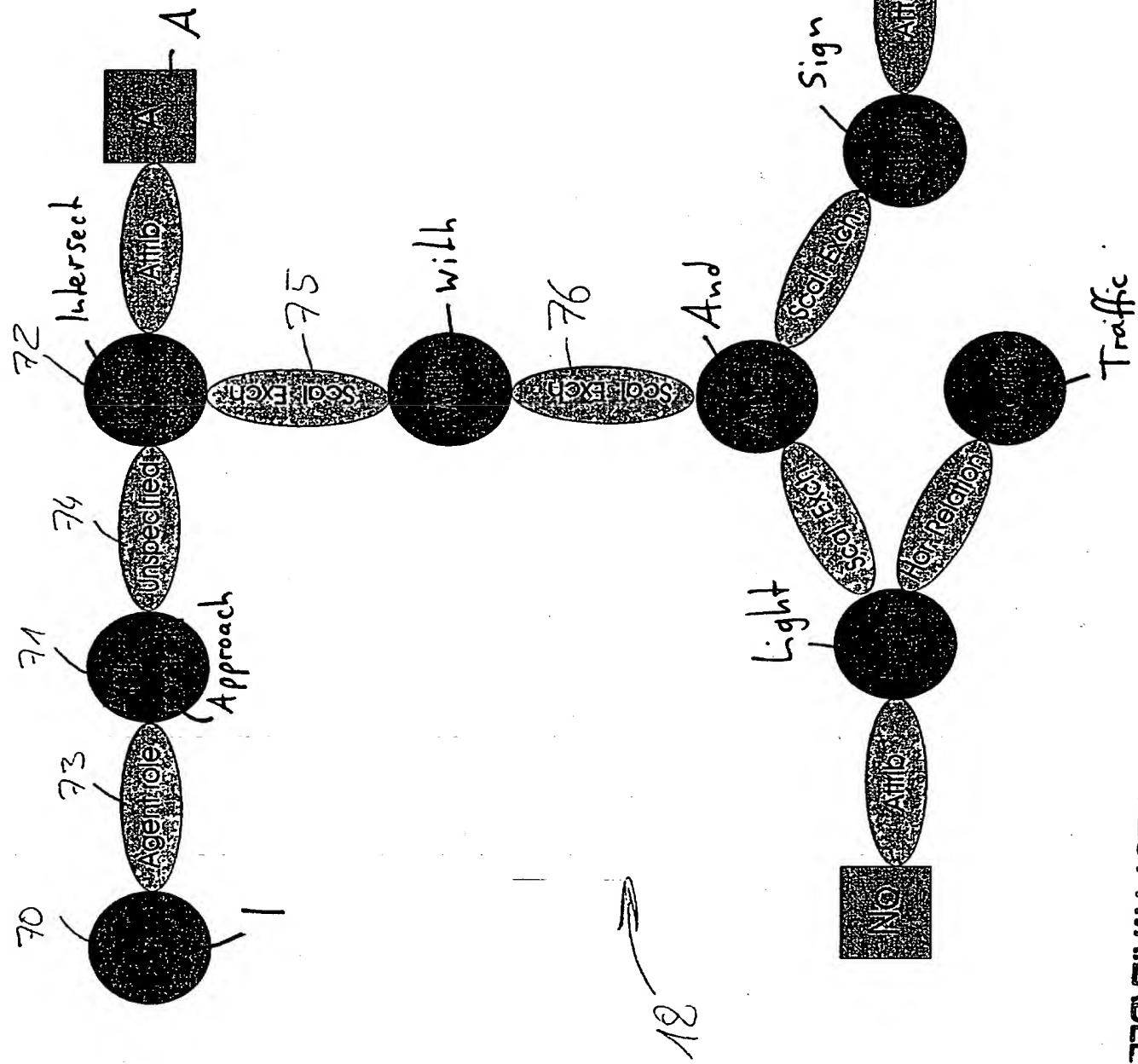
9/10

SEMANTIC   UNIT DECISIONS:

0-element of the Syntax Type String = 'NOUN'   ---> object 'INFORMATION'
0-element of the Syntax Type String = 'VERB'   ---> object 'INFORMATION'
0-element of the Syntax Type String = 'SUBCONJ' ---> object 'INFORMATION'
0-element of the Syntax Type String = 'PREP'   ---> object 'INFORMATION'
0-element of the Syntax Type String = 'DET'   ---> object 'ATTRIBUTE'
0-element of the Syntax Type String = 'ADV'   ---> object 'ATTRIBUTE'
0-element of the Syntax Type String = 'ADJ'   ---> object 'ATTRIBUTE'
0-element of the Syntax Type String = 'INFTO'   ---> object 'ATTRIBUTE'
0-element of the Syntax Type String = 'QUAL'   -- object 'Information

60

61   62

CONNECTION DECISIONS:

1) SyntaxTypeString (element 0) = VERB
   Slot (element 0) = the other word's position
   OtherSyntaxTypeString (element 0) = NOUN

   -> Create agentrole connection (type 0)

2) SyntaxTypeString (element 0) = VERB
   Slot (element 1) = the other word's position
   OtherSyntaxTypeString (element 0) = NOUN
   No passive form
   No gerund form

   -> Create objectrole connection (type 1)

3) other words entry type is 2

   -> Create attribute connection

4) SyntaxTypeString (element C) = NOUN
   OtherWord = WITH

   -> Create scaling exchange connection

5) OtherSemanticalTypeString = NNOUN

63

Fig. 6A

SyntaxTypeString (element 0) = NOUN

-> Create HorizontalRelation

6) word = WHILE
    otherSyntaxTypeString (element 0) = VERB

-> Create horizontal relation (type 12)

63

7) otherWord = WHILE
    syntaxTypeString (element 0) = VERB

-> Create horizontal relation (type 12)

8) syntaxTypeString(element 0) = VERB
    otherSyntaxTypeString (element 0) = VERB
    Gerund form

-> Create agent role connection (type 0)

9) SyntaxTypeString(element 0) = VERB
    Slot (element 2) = the other word's position
    OtherSyntaxTypeString(element 0) = NOUN
    No passive form

-> Create beneficiary role connection (type 7) (indirect object)

Fig. 6B